An Introduction to C#, Additional Documentation

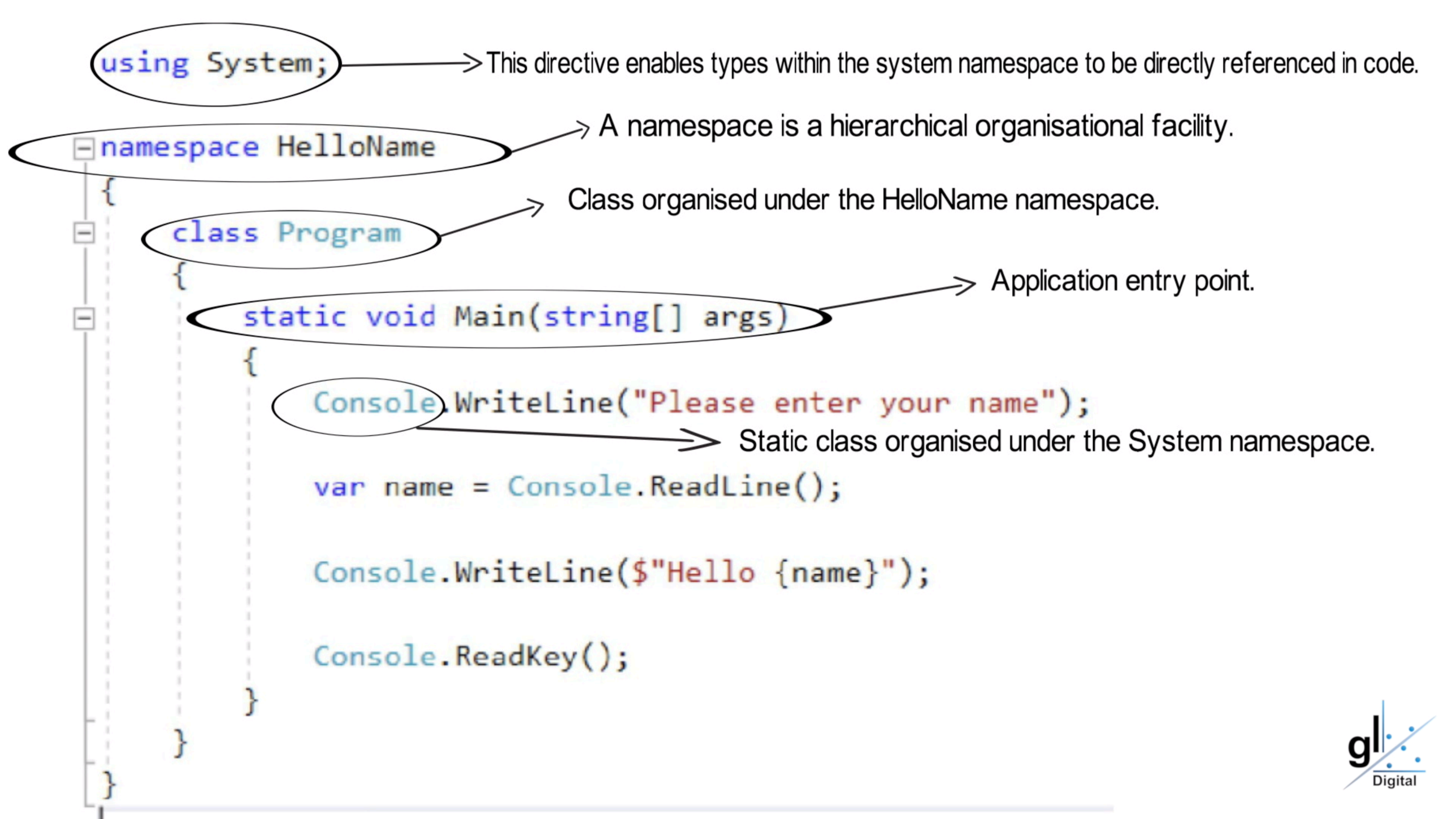
Introduction to C#

C# for Beginners

gl Digital

Main Method

- The Main method is the entry point of a C# application; it is where the program control starts and ends (note that the exceptions to this are library projects and windows services projects which don’t require a Main method entry point).
- There can only be one entry point for a c# application.
- Main is declared inside a class or struct.
- Main must be static. The enclosing class or struct is not required to be static.
- Main can either have a void or int return type.
- The Main method can be declared with or without a string[] array parameter that contains command-line arguments.


using System;
This directive enables types within the system namespace to be directly referenced in code.
namespace HelloName
A namespace is a hierarchical organisational facility.
class Program
Class organised under the HelloName namespace.
static void Main(string[] args)
Application entry point.
Console.WriteLine("Please enter your name");
Static class organised under the System namespace.
var name = Console.ReadLine();
Console.WriteLine($"Hello {name}");
Console.ReadKey();
}
}
}

c# is case sensative.

```
using System;

namespace HelloName
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Please enter your name");

            var Name = Console.ReadLine();

            Console.WriteLine($"Hello {name}");

            Console.ReadKey();
        }
    }
}
```

The name 'name' does not exist in the current context

Show potential fixes (Alt+Enter or Ctrl+.)

gl Digital

C# Code

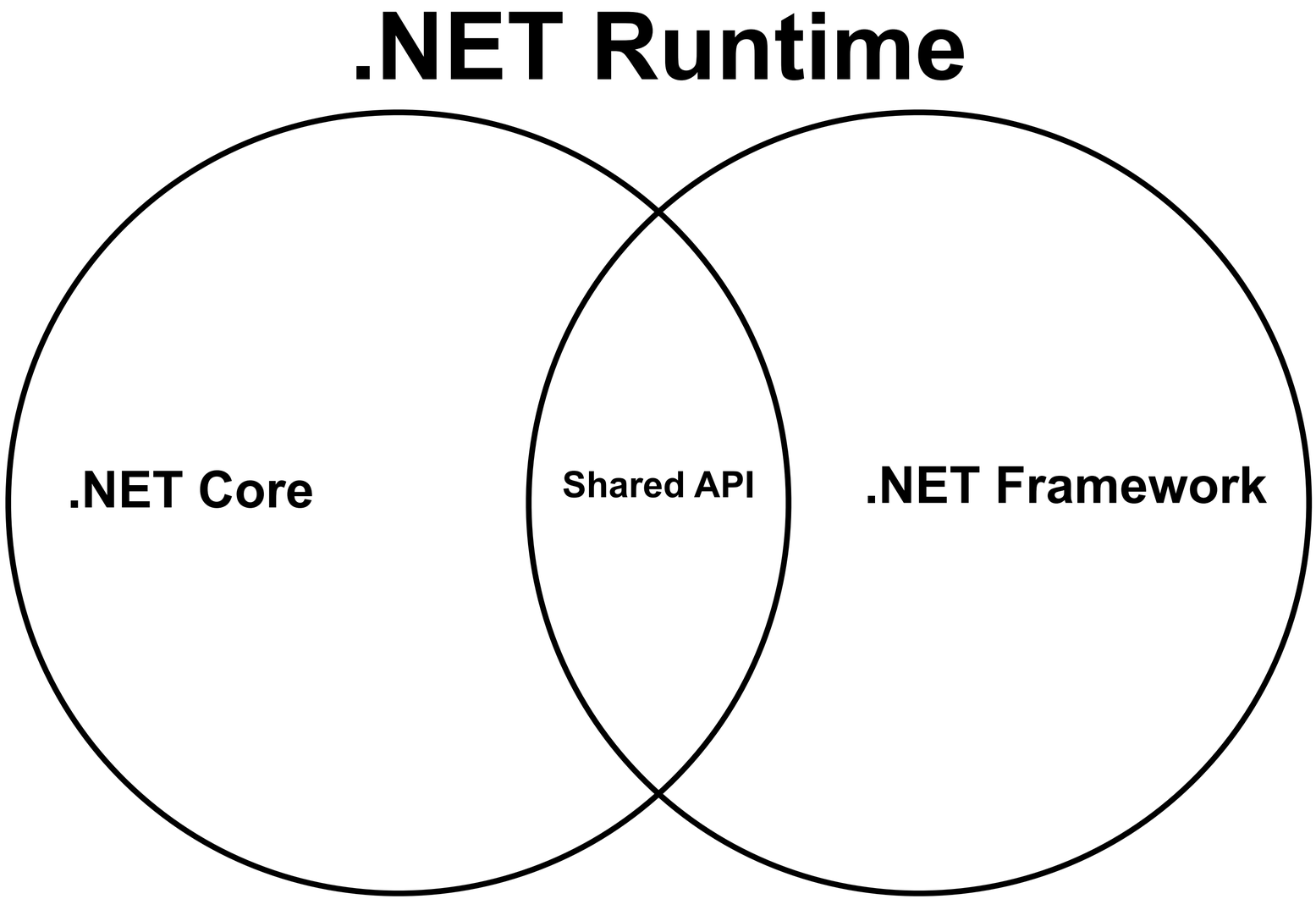
```csharp
using System;

namespace HelloName
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Please enter your name");

            var name = Console.ReadLine();

            Console.WriteLine($"Hello {name}");

            Console.ReadKey();
        }
    }
}
```

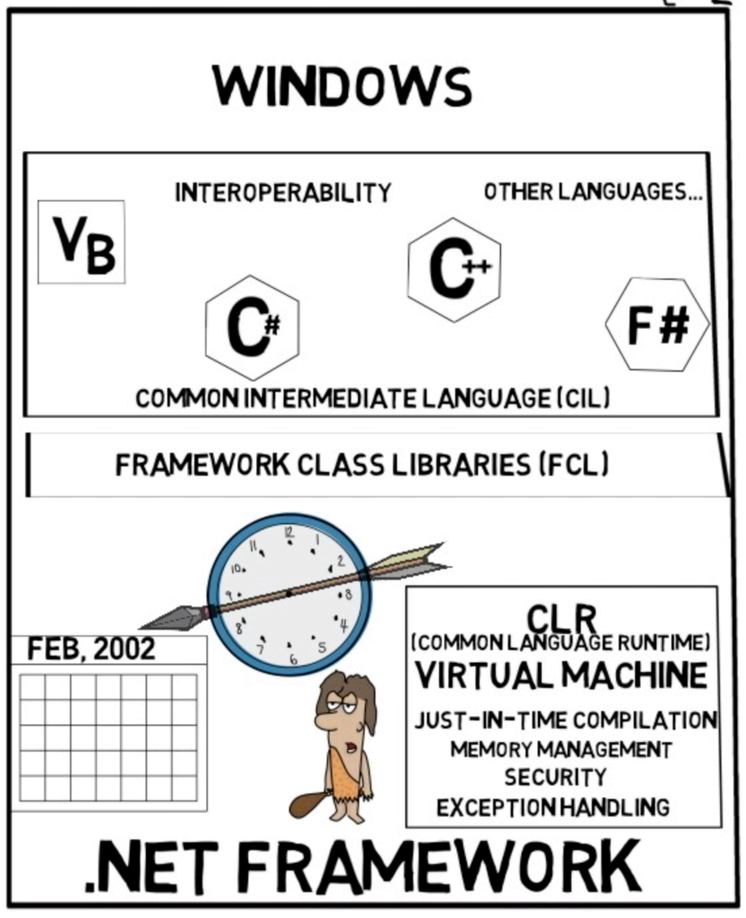
VB Code

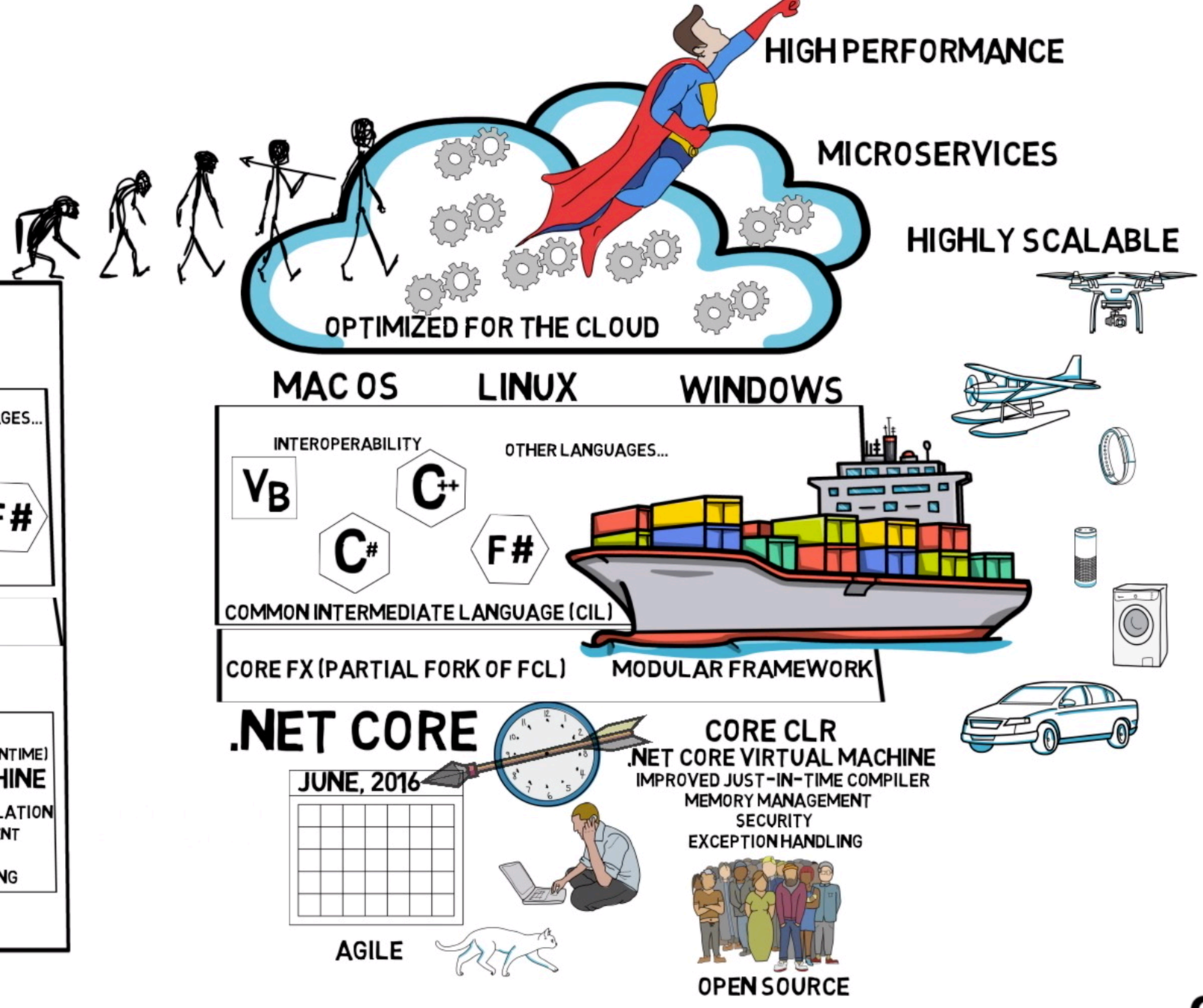
VB is not case sensative.

```vb
Imports System

Module Program

    Sub Main(args As String())

        Console.WriteLine("Please enter your name")

        Dim Name = Console.ReadLine()

        Console.WriteLine($"Hello {name}")

        Console.ReadKey()

    End Sub

End Module
```

gl Digital

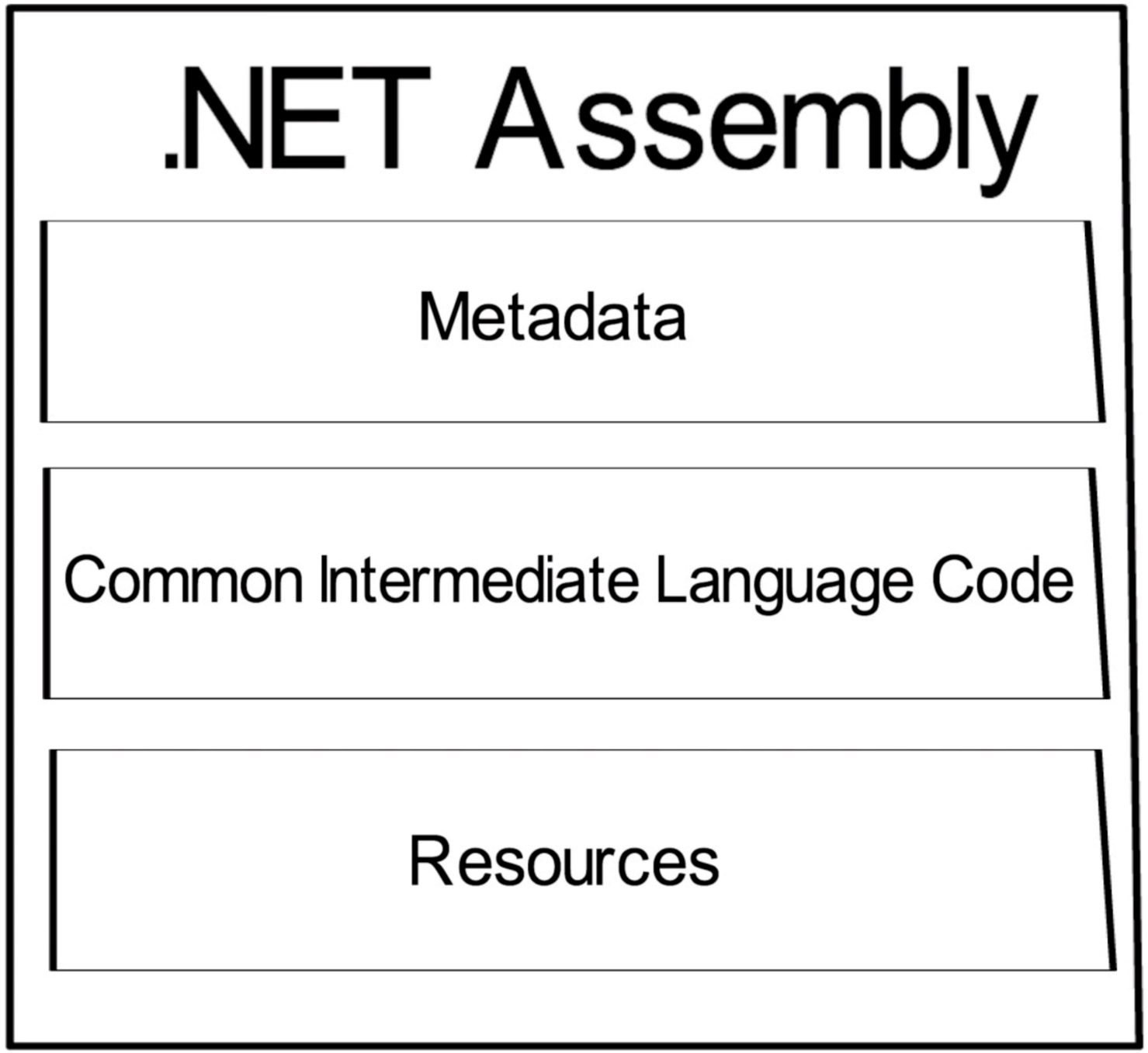

.NET Runtime
.NET Core
Shared API
.NET Framework

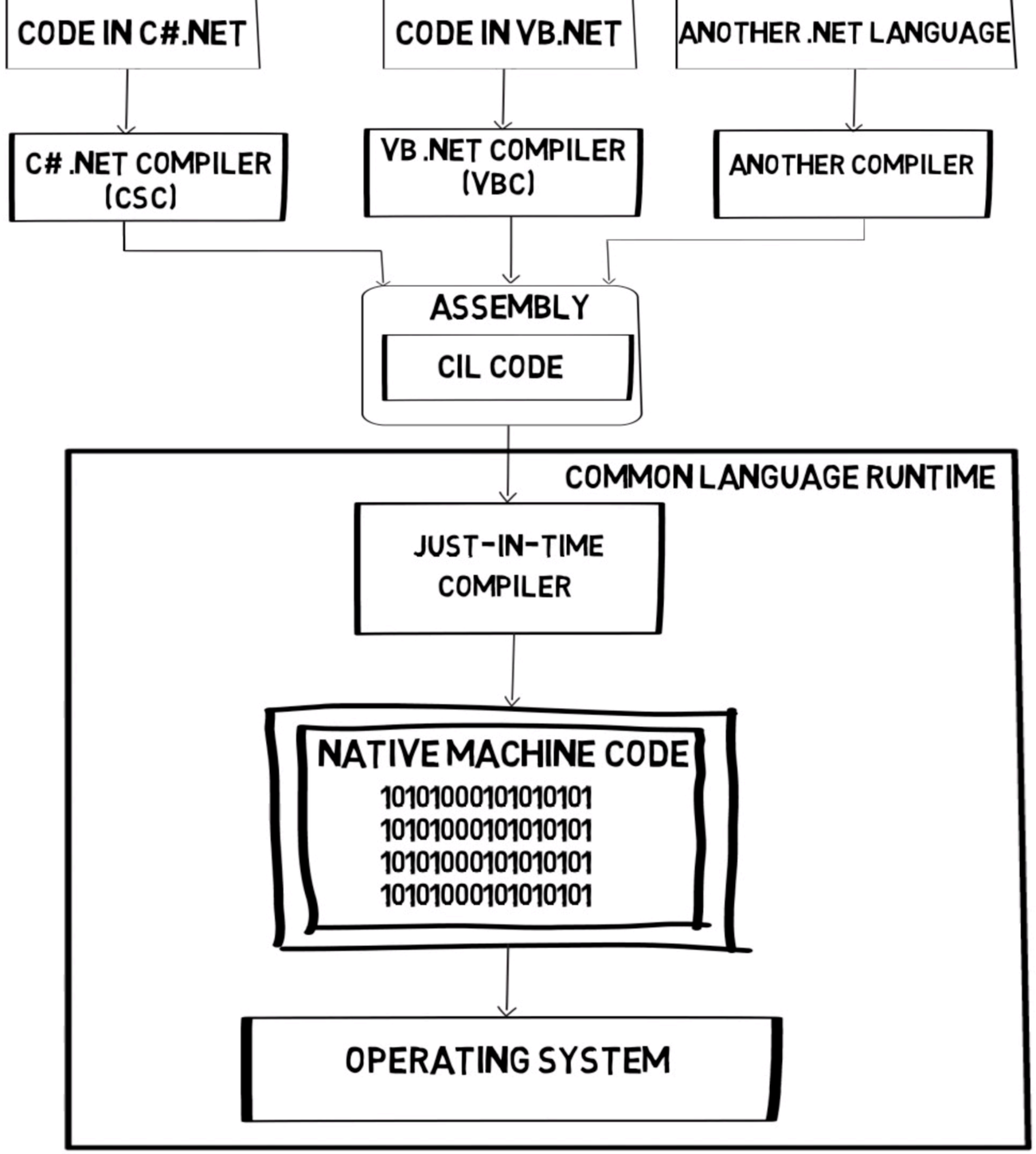

WINDOWS
INTEROPERABILITY
OTHER LANGUAGES...
VB
C++
C#
F#
COMMON INTERMEDIATE LANGUAGE (CIL)
FRAMEWORK CLASS LIBRARIES (FCL)
FEB, 2002
CLR
(COMMON LANGUAGE RUNTIME)
VIRTUAL MACHINE
JUST-IN-TIME COMPILATION
MEMORY MANAGEMENT
SECURITY
EXCEPTION HANDLING
.NET FRAMEWORK


HIGH PERFORMANCE
MICROSERVICES
HIGHLY SCALABLE
OPTIMIZED FOR THE CLOUD
MAC OS
LINUX
WINDOWS
INTEROPERABILITY
OTHER LANGUAGES...
VB
C++
C#
F#
COMMON INTERMEDIATE LANGUAGE (CIL)
CORE FX (PARTIAL FORK OF FCL)
MODULAR FRAMEWORK
.NET CORE
JUNE, 2016
AGILE
CORE CLR
.NET CORE VIRTUAL MACHINE
IMPROVED JUST-IN-TIME COMPILER
MEMORY MANAGEMENT
SECURITY
EXCEPTION HANDLING
OPEN SOURCE

.NET Assembly

- **Assemblies** are the **building blocks** of **.NET Framework** applications.
- They form the **fundamental unit of deployment**, version control, reuse, activation scoping, and security permissions.
- An **assembly** is a **collection** of **types** and **resources** that are built to work together and form a **logical unit of functionality**.
- An assembly provides the **Common Language Runtime** with the information it needs to be aware of **type implementations**.
- To the runtime, a type does not exist outside the context of an assembly.
- A .NET assembly can be an **executable** file (with a .exe file extension) or a **dynamic linked library** file (with a .dll file extension).


.NET Assembly
Metadata
Common Intermediate Language Code
Resources


CODE IN C#.NET
CODE IN VB.NET
ANOTHER .NET LANGUAGE
C# .NET COMPILER (CSC)
VB .NET COMPILER (VBC)
ANOTHER COMPILER
ASSEMBLY
CIL CODE
COMMON LANGUAGE RUNTIME
JUST-IN-TIME COMPILER
NATIVE MACHINE CODE
10101000101010101
10101000101010101
10101000101010101
10101000101010101
OPERATING SYSTEM

Summary

- We have written a very basic C# application and discussed some of the important elements pertaining to our C# code.
- We coded this application using four different code editors to demonstrate that our .NET code is not tightly coupled to any one particular code editor.
- We also ran our .NET code on a Windows Operating System as well as a macOS platform.
- We discussed that our code was cross platform because we were using .NET Core as our runtime environment and that this would not have been possible using the .NET Framework.
- We broadly looked at the .NET Framework runtime and its more evolved counter part, .NET Core.
- We briefly looked at .NET assemblies which are the fundamental building blocks of .NET applications.
- We also briefly discussed how our code is processed by the .NET runtime.


gl
Digital